PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-260714
(43)Date of publication of application : 29.09.1998

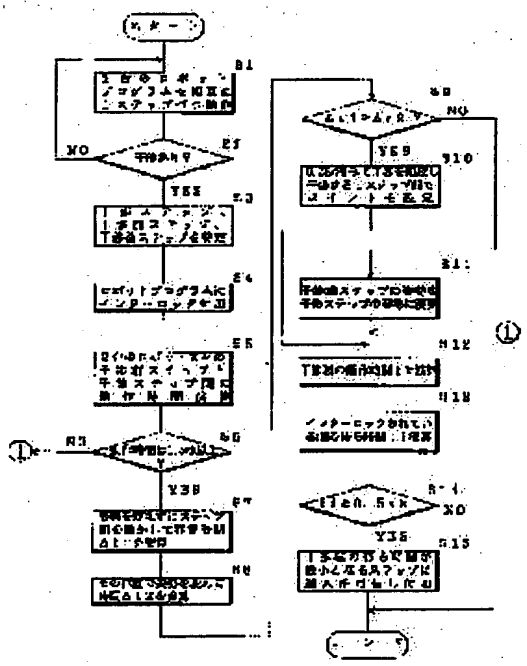
(51)Int.CI. G05B 19/4093
B25J 19/06
G05B 9/02
G05B 11/32
G05B 19/18

(21)Application number : 09-068280
(22)Date of filing : 21.03.1997

(71)Applicant : NISSAN MOTOR CO LTD
(72)Inventor : SHIMAMURA SHIGEO
IWASA TATSUKI

(54) ROBOT INTERFERENCE AREA SETTING PROGRAM PREPARING METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To automatically prepare the program of optimized robot.
SOLUTION: The plural robots are operated while advancing respective programs step by step, the steps of operating area to be mutually interferred with the other robot are found for each robot (S1 and S2) and even when that robot enters the relevant mutual interference operation area, advance permission information permitting the advancement into the relevant mutual interference operation area of the other robot is automatically set to the program of the other robot interferring with that robot (S11-S15).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－260714

(43)公開日　平成10年(1998)９月29日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０５Ｂ　19/4093		Ｇ０５Ｂ　19/403	Ｅ
Ｂ２５Ｊ　19/06		Ｂ２５Ｊ　19/06	
Ｇ０５Ｂ　9/02		Ｇ０５Ｂ　9/02	Ｇ
11/32		11/32	Ａ
19/18		19/18	Ｃ

審査請求　未請求　請求項の数４　ＯＬ　（全 11 頁）

(21)出願番号　特願平9－68280

(22)出願日　平成９年(1997)３月21日

(71)出願人　000003997
日産自動車株式会社
神奈川県横浜市神奈川区宝町２番地

(72)発明者　嶋村　繁生
神奈川県横浜市神奈川区宝町２番地　日産自動車株式会社内

(72)発明者　岩佐　達樹
神奈川県横浜市神奈川区宝町２番地　日産自動車株式会社内

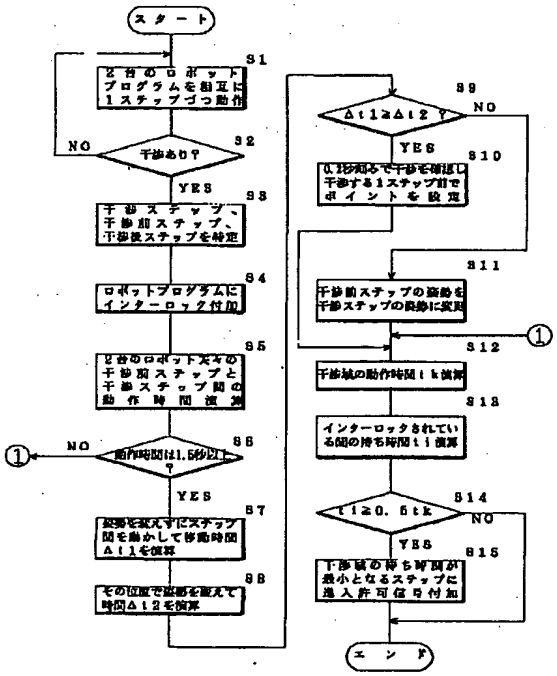
(74)代理人　弁理士　八田　幹雄　（外１名）

(54)【発明の名称】　ロボット干渉域設定プログラム作成方法

(57)【要約】

【課題】　最適化されたロボットのプログラムを自動的に作成する。

【解決手段】　複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め（Ｓ１，Ｓ２）、そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定する（Ｓ１１～Ｓ１５）。



1

【特許請求の範囲】
【請求項１】　互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、
当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、
そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法。
【請求項２】　互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、
当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、
当該干渉し合う動作領域を分割して複数の干渉域を設定し、
そのロボットが他のロボットと干渉し合う動作領域に入り、第ｎ干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法。
【請求項３】　互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、
当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、
当該干渉し合う動作領域を分割して複数の干渉域を設定し、
そのロボットが他のロボットと干渉し合う動作領域に入るそのロボットのプログラムの最初のステップに、他のロボットが既に当該干渉し合う動作領域に入っている場合にそのロボットの干渉を回避させるためのインターロック情報を付加し、
そのロボットが他のロボットと干渉し合う動作領域に入り、第ｎ干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに付加し、

2

また、そのロボットが他のロボットと干渉し合う動作領域から脱出するそのロボットのプログラムの最初のステップに、他のロボットが当該干渉し合う動作領域に入ることを許すインターロック解除情報を付加することを特徴とするロボット干渉域設定プログラム作成方法。
【請求項４】　前記進入許可情報の付加は、この許可情報を任意のロボットのプログラムに付加した場合に、この許可情報が付加されたロボットがこの許可情報が付加された干渉域の動作に要する時間と他のロボットがこのロボットの干渉域の動作のために待たされる時間とを勘案し、他のロボットのその干渉域への進入待ち時間が最短となるようなステップに付加されることを特徴とする請求項２または請求項３に記載のロボット干渉域設定プログラム作成方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、互いに干渉し合う動作領域を共有する複数のロボットのプログラムに、他のロボットとの干渉を回避させるための情報を自動的に付加したり、干渉を回避する待ち時間を最小にする情報を自動的に付加したりすることができるロボット干渉域設定プログラム作成方法に関する。
【０００２】
【従来の技術】車体スポット溶接工程のように、ワーク周辺に複数台のロボットが密接して設置してある設備の場合には、ロボット相互の干渉を避けるために、ロボット間の動作の優先付けをし、干渉域に後から進入するロボットは、先に干渉域に入った相手のロボットが干渉域を出るまで、干渉域に入る１ステップ前で待機するようにプログラムされる。従来は、このプログラムの作成を、実機を操作して干渉を逐一確認しながら行っている。
【０００３】実機を用いて干渉回避のプログラムを作成する手法を２台のロボットの場合について述べる。まず、それぞれのロボットにオペレータが付き、ロボットのプログラムを片方のロボットについて１ステップづつ動作させておおまかな干渉域を把握する。つぎに、両方のロボットを衝突させないように部分的なプレイバックをしながら干渉域のステップを探す。つぎに、どちらのロボットが先に干渉域に入るのかを時間を測定して調べる。最後に、干渉域に入る１ステップ前と１ステップ後に干渉域進入信号および干渉域外信号を設定する。
【０００４】
【発明が解決しようとする課題】このような方法で干渉回避のプログラムを作成したのでは、実機の衝突の危険を常に孕んでおり、また、トライアンドエラーによるプログラムの作成となるので、その作成には非常に多くの時間を必要とする。さらに、最適なプログラムの作成（どの位置でどの姿勢で退避させるのか）には熟練を要する。

【０００５】また、最近では、グラフィックシミュレータ等を用い、いわゆるオフラインティーチングを行うことによって干渉回避のプログラムを作成することもできるようになったが、干渉域の確認時に実機の衝突のリスクを避けることができるとはいうものの、プログラムの作成が試行錯誤であるということは実機を用いた場合と同じである。

【０００６】したがって、干渉をも考慮しなければならないロボットのプログラムの作成には非常に多くの時間を要するという問題と、サイクルタイムを最短とするプログラムの作成を容易に行うことができない（熟練を要する）という問題とがある。

【０００７】本発明は、以上のような問題点を解消すべく成されたものであり、互いに干渉し合う動作領域を共有する複数のロボットのプログラムに、他のロボットとの干渉を回避させるための情報を自動的に付加したり、干渉を回避する待ち時間を最小にする情報を自動的に付加したりすることができ、複数のロボットを待ち時間を最小としつつ最小のサイクルタイムで動作させることができるロボット干渉域設定プログラム作成方法を提供することを目的とする。

【０００８】

【課題を解決するための手段】上記目的を達成するための本発明は、次のように構成される。請求項１に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法である。

【０００９】このような方法をとれば、他方のロボットが干渉し合う動作領域を出るまで一方のロボットを待たせておかなくとも良くなり、ロボットのサイクルタイムを短縮化することができるようになる。

【００１０】請求項２に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定

し、そのロボットが他のロボットと干渉し合う動作領域に入り、第ｎ干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法である。

【００１１】このような方法をとれば、他方のロボットが干渉し合う動作領域を出るまで一方のロボットを待たせておかなくとも良くなり、またこの領域に入った場合でもロボット同志の干渉を確実に回避しながら、ロボットのサイクルタイムを短縮化することができるようになる。

【００１２】請求項３に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを１ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定し、そのロボットが他のロボットと干渉し合う動作領域に入るそのロボットのプログラムの最初のステップに、他のロボットが既に当該干渉し合う動作領域に入っている場合にそのロボットの干渉を回避させるためのインターロック情報を付加し、そのロボットが他のロボットと干渉し合う動作領域に入り、第ｎ干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに付加し、また、そのロボットが他のロボットと干渉し合う動作領域から脱出するそのロボットのプログラムの最初のステップに、他のロボットが当該干渉し合う動作領域に入ることを許すインターロック解除情報を付加することを特徴とするロボット干渉域設定プログラム作成方法である。

【００１３】このような方法によれば、複数のロボットを効率的に動作させることができるようになり、結果として、ロボットのサイクルタイムを短縮化でき、作業効率が向上することになる。

【００１４】請求項４に記載の発明は、請求項２または請求項３に記載の発明において、前記進入許可情報の付加は、この許可情報を任意のロボットのプログラムに付加した場合に、この許可情報が付加されたロボットがこの許可情報が付加された干渉域の動作に要する時間と他のロボットがこのロボットの干渉域の動作のために待たされる時間とを勘案し、他のロボットのその干渉域への進入待ち時間が最短となるようなステップに付加されることを特徴とするロボット干渉域設定プログラム作成方

法である。

【００１５】このように最適なステップに進入許可情報を自動的に付加できるようにすれば、短時間で各ロボットのプログラムを最適化することができるようになり、プログラムの作成時間を短縮することができるようになる。

【００１６】

【発明の効果】本発明は、次のような効果を奏する。請求項１から請求項３の発明では、そのロボットと干渉する他のロボットを、干渉し合う動作領域内であっても、干渉しない範囲内で動作させることができるロボットプログラムを容易に作成することができ、プログラムの作成時間を大幅に短縮することができる。また、この作成はオペレータがいなくても自動的に行われるので、プログラムの作成に携わる人員の削減をすることもできる。

【００１７】請求項４の発明では、進入許可情報を付加する最適なステップを自動的に捜し出せるので、熟練を要することなく、だれでも短時間の作業で最適なロボットプログラムの作成が可能となる。

【００１８】

【発明の実施の形態】以下に、本発明のロボット干渉域設定プログラム作成方法について説明する。図１は、本発明方法を用いてプログラムされる２台のロボットＡ，Ｂを示してある。これらのロボットは、それぞれが図示されているような軌跡を描くようにプログラムされている。このようにプログラムされているロボットを他のロボットとは無関係に独立して動かしたのでは、当然のことながらアーム同志が衝突（干渉するステップは黒丸で、干渉しないステップは白丸で示す）してしまう。これを回避させながら効率良く動かせるようなプログラムを自動的に作成できるようにするのが本発明である。特に、干渉する領域内でも、お互いに干渉しない範囲内でできるだけ双方のロボットが待ち時間なく動作できるようにするプログラムを自動的に作成できるようにしている点が本発明では特筆される。

【００１９】本発明の方法を実施する装置は、図２に示すような構成となっている。図中、ロボットプログラム記憶部１０は、図１のようにそのロボットの動きのみが書き込まれているプログラムを外部から読み込んで記憶させておくものである。たとえば、ロボットＡのプログラムはこの記憶部１０のあるエリアに、また、ロボットＢのプログラムは記憶部１０の他のエリアに記憶される。

【００２０】シミュレーション実行部１２は、ロボットプログラム記憶部１０からそれぞれのロボットＡ，Ｂのプログラムを読み込んで、画面上でロボットＡ，Ｂを交互に１ステップづつ動かす機能を有するものである。また、各ステップごとの動作時間を計測する機能も持っている。

【００２１】干渉チェック部１４は、シミュレーション実行部１２によってロボットＡ，Ｂを動かした結果、両ロボットＡ，Ｂが「干渉を起こすステップ」がどのステップからどのステップであるのかを、それぞれのロボットＡ，Ｂについて認識することができるものであり、干渉する前のステップを干渉前ステップとして、干渉するステップを干渉ステップとしてそれぞれ設定したり、サイクルタイムの最適化を図るための新たなステップを作り出したりするための諸機能も有している。

【００２２】インターロック自動設定部１６は、干渉チェック部１４によって認識された「干渉を起こすステップ」に基づいて、両ロボットＡ，Ｂのステップにインターロック情報やインターロック解除情報を自動的に付加する指令を出力することができるものである。また、干渉チェック部１４における新たなステップの付加指令を受けてそれを出力することもできるものである。さらに、動作時間演算部２０の演算結果に基づいて最適なステップに進入許可情報を付加する指令を出力することができるものである。

【００２３】ロボットプログラム変換部１８は、インターロック自動設定部１６からのインターロック情報やインターロック解除情報を付加すべき指令や新たなステップの付加指令、さらには進入許可情報を付加する指令に基づいて、ロボットＡ，Ｂの各プログラムにそれらの情報を付加したり、新たなステップを付加したりするために、ロボットプログラム記憶部１０に記憶されているそれぞれのロボットＡ，Ｂのプログラムをそれらの指令にしたがって変換し、出力することができるものである。

【００２４】動作時間演算部２０は、ロボットがお互いに干渉し合う動作領域（図１の黒丸の領域）を複数の干渉域に分割し、任意のステップに仮想的に進入許可情報を付加した場合に、それぞれの干渉域の動作にそれぞれのロボットがどの位の時間がかかるのかを演算し、どのステップに進入許可信号を付加した場合がそれぞれのロボットの待ち時間が最小となってサイクルタイムが最小となるかをみつけ出し、そのステップに進入許可信号を付加すべき指令を出力するものである。

【００２５】つぎに、本発明にかかるロボット干渉域設定プログラム作成方法の具体的な手順を図３に示したフローチャートに基づいて説明する。まず、ロボットプログラム記憶部１０は、図１に示した２台のロボットＡ，Ｂのそれぞれのプログラムをそれぞれのエリアに記憶する。そして、シミュレーション実行部１２では、この記憶部１０に記憶されているそれぞれのロボットのプログラムを１ステップづつ読み出して、２台のロボットを交互に動作させる。この処理によって２台のロボットは画面上で動くことになり、軌跡は例えば図４に示すようになる（Ｓ１）。干渉チェック部１４では、この２台のロボットが干渉していないかを常に監視し、干渉が検出された場合には、干渉を起こす前のプログラムのステップを干渉前ステップに、干渉を起こしたステップを干渉ス

テップに、干渉した後のステップを干渉後ステップとする。このステップの特定は、２台のロボットのそれぞれのプログラムに対して行われる。

【００２６】たとえば図４に示すように、ロボットＡは、ステップａがロボットＢとの干渉を起こす直前のステップであるのでこれを干渉前ステップとし、ステップｂに歩進させるとロボットＢとの干渉を起こすので、これを干渉ステップとし、ｃを干渉後ステップとする。また、ロボットＢについては、ステップａ′を干渉前ステップに、ステップｂ′を干渉ステップに、ステップｃ′を干渉後ステップにそれぞれ特定する。

【００２７】なお、これらのステップの特定は、２台のロボットを最初のステップから最後のステップまで動かしてみてから行っても良いし、干渉と同時にリアルタイムで行っても良い（Ｓ２，Ｓ３）。

【００２８】以上の処理によって、ロボットＡは、そのプログラムのどのステップからどのステップまでの処理においてロボットＢと干渉する恐れがあるのか、換言すれば、干渉し合う動作領域がどのステップからどのステップであるのかがわかる。同様にしてロボットＢについてもロボットＡと干渉し合う動作領域がどのステップからどのステップであるのかがわかる。

【００２９】つぎに、インターロック自動設定部１６では、干渉チェック部１４によって特定されたそれぞれのロボットのプログラムの干渉前ステップ（ａ，ａ′）にインターロック情報を付加すべき指令を出し、また、干渉し合う動作領域を脱出した直後のステップ（ｄ，ｄ′）にインターロック解除情報を付加すべき指令を出す。これらの情報の付加は、それぞれのステップに対してしても良いし、新たなステップをこれらのステップの前、または後に設けても良い。これらの情報の付加は、インターロック自動設定部１６の指令によりロボットプログラム変換部１８が行う。すなわち、インターロック自動設定部１６からの指令に基づいて、ロボットプログラム記憶部１０からそれぞれのロボットＡ，Ｂのプログラムを読み込み、それぞのプログラムの該当するステップにインターロック情報、インターロック解除情報を付加し、これを新ロボットプログラムとして外部に出力する。

【００３０】たとえば、ロボットＡのプログラムの干渉ステップａに、ロボットＢが干渉し合う動作領域に既に入っている場合には、ロボットＢがこの動作領域を出るまで、換言すればロボットＢのステップｄ′の処理によってロボットＡに対してインターロック解除信号を出力するまで，そのステップで退避させるインターロック情報を付加する。また、ロボットＢのプログラムの干渉ステップａに、ロボットＡが干渉し合う動作領域に既に入っている場合には、ロボットＡがこの動作領域を出るまで、換言すればロボットＡのステップｄの処理によってロボットＢに対してインターロック解除信号を出力するまで，そのステップで退避させるインターロック情報を付加する。さらに、ロボットＡのプログラムのステップｄには、ロボットＢを自由に動作させるためのインターロック解除信号を、ロボットＢのプログラムのステップｄ′には、ロボットＢを自由に動作させるためのインターロック解除信号をそれぞれ付加する（Ｓ４）。

【００３１】さらに、干渉チェック部１４では、ロボットＡ、ロボットＢのそれぞれのプログラムに特定された干渉前ステップと干渉ステップとに基づいて、干渉前ステップから干渉ステップに移行するまでに要する時間、すなわち動作時間を演算し、この動作時間が基準時間（この実施形態では１．５秒を想定している）よりも小さければ、ロボットのサイクルタイムのこれ以上の短縮化はできないものと判断して処理を終了する（Ｓ５，Ｓ６）。

【００３２】一方、この動作時間が基準時間よりも大きければ、ロボットの退避位置を干渉し合う動作領域側に移動させることによって、また、姿勢を事前に変化させておくことによってロボットのサイクルタイムの短縮をすることができるようになるので、この場合には次の処理をする。

【００３３】まず、干渉チェック部１４が干渉前ステップから干渉ステップへの動作時間が１．５秒以上であると判断したときには、シミュレーション実行部１２に、ロボットの姿勢（ｐ，ｑ，ｒ）を変えずに干渉前ステップ（ｘ１，ｙ１，ｚ１）から干渉ステップ（ｘ２，ｙ２，ｚ２）までの移動を指示し、干渉チェック部１４でこの間の移動時間Δｔ１を演算する。つまり、ステップ間の平行移動に要する時間を演算する（Ｓ７）。

【００３４】そしてつぎに、シミュレーション実行部１２に、ロボットの位置（ｘ，ｙ，ｚ）を変えずに姿勢のみを変化させる指示をし、干渉チェック部１４でロボットの位置を干渉前ステップから動かさずに、ロボットの姿勢を干渉前ステップの姿勢（ｐ１，ｑ１，ｒ１）から干渉ステップの姿勢（ｐ２，ｑ２，ｒ２）に変化させるに要する時間Δｔ２を演算する（Ｓ８）。

【００３５】干渉チェック部１４では、この演算したΔｔ１とΔｔ２の時間を比較し、Δｔ１≧Δｔ２であれば、姿勢を変えるよりもステップ間の移動に時間を要しているのであるから移動時間を少しでも短縮することができればロボットの全体のサイクルタイムの短縮化を図ることができるようになるので、干渉前ステップの位置を干渉し合う動作領域側に移動させるようにする。つまり、ロボットが干渉の退避をする位置を干渉ステップ側に移動させる。この退避位置をどこに設けるかは、図５に示すように、時間にして０．２秒分だけ退避位置ａを干渉し合う動作領域側（ｂ側）に移動させ（図の軌跡の１刻みに相当）、この点での干渉の確認をし、干渉しなければさらに０．２秒分動かすという操作を繰り返し行い、干渉が確認された点の１つ前の点ａ１を新たな退避

位置とする。干渉チェック部１４は、この新たな退避位置を設けるために、インターロック自動設定部１６にこの点に新たなステップを設けるべきことを指令し、ロボットプログラム変換部１８はインターロック自動設定部１６からのこの指令に基づいて、ロボットプログラム記憶部１０から指定のロボットのプログラムを読み込み、そのプログラムの干渉前ステップと干渉ステップとの間に新たなステップａ１（図５参照）を挿入する処理をする（Ｓ９，Ｓ１０）。

【００３６】一方、Δｔ１＜Δｔ２であれば、位置を変えるよりもステップ間の姿勢の変化に時間を要しているのであるから姿勢を事前に干渉ステップの姿勢にしておけばロボットの全体のサイクルタイムの短縮化を図ることができるようになるので、干渉前ステップの姿勢を干渉ステップの姿勢にするようにする。つまり、ロボットが干渉の退避をしている間に、次のステップ（干渉ステップ）の姿勢に備えておく。

【００３７】干渉チェック部１４は、干渉前ステップのロボットの姿勢を干渉ステップのロボットの姿勢に変更させるために、インターロック自動設定部１６に姿勢変更の指令を出力し、ロボットプログラム変換部１８は、インターロック自動設定部１６からのこの指令に基づいて、ロボットプログラム記憶部１０から指定のロボットのプログラムを読み込み、このプログラムの干渉前ステップの姿勢を干渉ステップの姿勢に変更する処理をする（Ｓ１１）。

【００３８】次に、動作時間演算部２０は、ロボットＡとロボットＢのそれぞれに対して干渉し合う動作領域を動作させるのにかかる時間ｔｋを演算する（Ｓ１２）。また、一方のロボットがこの干渉し合う動作領域に入っている場合に、他方のロボットの待ち時間、すなわちインターロックされている間の動作時間ｔｉを演算する。これは、両方のロボットを動かした場合の一方のロボットの待ち時間がどの程度の時間となるのかを調べるためである（Ｓ１３）。

【００３９】この待ち時間ｔｉが動作時間ｔｋの５０％よりも大きい時間であれば、一方のロボットは他方のロボットがこの干渉し合う動作領域から脱出するまでずっと待ち続けなければならないので、サイクルタイムが必然的に大きくなり、ロボットのプログラムとしてはあまり好ましいものではない。したがって、この場合には、ロボットの待ち時間ができるだけ少なくなるように、一方のロボットが干渉し合う動作領域に入っている場合であっても、干渉しないステップまでは他のロボットの動作を許すような進入許可信号を付加する。この進入許可信号を付加する指令は動作時間演算部２０が出す。なお、どのステップに進入許可信号を付加するかの最適化の演算は後述する（Ｓ１４，Ｓ１５）。

【００４０】一方、この待ち時間ｔｉが動作時間ｔｋの５０％よりも小さい時間であれば、許容できる範囲の待ちの発生であると判断してプログラムの作成を終了する（Ｓ１４）。

【００４１】なお、進入許可信号をプログラム中に付加するか付加しないかの判断は、待ち時間ｔｉが動作時間ｔｋの５０％よりも大きいか小さいかによって行っているが、判断の基準となる割合は、ロボットの動作によって任意に設定されるものである。たとえば、一方のロボットが他方のロボットに追従するような形でお互いの干渉域を動作するものであれば、その待ち時間の割合は１０％というような少ない値で良いと思うが、一方のロボットが任意の範囲で往復移動、この間他方のロボットを干渉域内に入れることができないような場合には、その待ち時間の割合は７０％というような大きな値が選ばれる。

【００４２】つぎに、進入許可信号の付加の方法について図６と図７を参照しながら詳細に説明する。この進入許可信号の付加は、次のようなことを可能とするために行われるものである。つまり、追従側のロボットは、図３に示したフローチャートのＳ１～Ｓ１１の処理だけでは、優先側のロボットが干渉し合う動作領域の処理を終わるまで待ち状態となってしまう。たとえば、どちらかのロボットが最初に干渉し合う動作領域に入った場合には、後からこの領域に入ろうとするロボットは、最初に入ったロボットがこの領域から出るまでの間、干渉前ステップでずっと待っていなければならなくなる。確かに、優先するロボットがこの領域を縦横無尽に動き回るってから出るような動きをするものであれば、後のロボットの待ち時間はけっして無駄な時間とはいえないが、優先するロボットが経路を戻ることなく順次進んで行き、後のロボットがこの優先するロボットの軌跡を追いかけて行くような場合には、優先するロボットがこの領域を脱出するまで後のロボットを待たせておくことは非常に無駄である。

【００４３】このような場合でも、後のロボットが干渉し合う動作領域内に入ることを許し、優先するロボットに干渉しない範囲内で後のロボットが追従して行けるようにすれば、後のロボットの待ち時間は少なくなるし、作業のサイクルタイムも短縮できる。Ｓ１１～Ｓ１５のステップでは、このようなことを可能としているのである。

【００４４】さて、図６および図７において、「優先」、「追従」とあるが、「優先」というのは、干渉し合う動作領域に最初に入ったロボットをいい、「追従」というのは、優先するロボットの動作を見ながら後から動作するロボットをいう。この例では、それぞれのロボットの干渉し合う動作領域をＡ，Ｂという２つの干渉域に分割している。図６に示すプログラムでは、優先するロボットのプログラムのステップ１１とステップ１２において、追従するロボットが干渉し合う動作領域（干渉域Ａ＋干渉域Ｂ）に入れないようにインターロック情報

（SOF4，SOF35 ）を設定している。したがって、優先するロボットがステップ１１，１２以降のステップの処理に進んでいる場合には、追従するロボットはこの動作領域に入ることができず、干渉前ステップまでの処理しかできない。すなわち、追従するロボットは干渉する領域に入る直前（ステップ２０のSOF4）で待機することになる。

【００４５】優先するロボットの処理が進み、ステップ２５の処理が終わると、追従するロボットが干渉域Ａに入ることが許され（SON ，４ ）る。このSON ，４ の情報は、干渉域Ａへの進入を許可する進入許可情報であるので、追従するロボットはこの情報により干渉域Ａに入ることができる。ただ、干渉域Ａでの動作が終了しても、干渉域Ｂには入ることができない。SOF35 というインターロック情報がまだ解除されていないからである（ステップ２９のSOF35 ）。

【００４６】さらに優先するロボットの処理が進み、ステップ２９の処理が終わると、追従するロボットが干渉域Ｂに入ることが許され（SON ，35）る。このSON ，35 の情報は、干渉域Ｂへの進入を許可する進入許可情報である。追従するロボットはこの情報により干渉域Ｂに入ることができるようになる。ここで、進入許可情報をどのステップに付加するかということがサイクルタイムを短縮する上で非常に重要となる。どのステップに付加するかによって、追従するロボットが待ち時間を少なくして干渉域に入ることができるかどうかが左右されるからである。

【００４７】この進入許可情報を付加するステップをどこにするかは、次のようにして決める。図７において、追従するロボットが最初のインターロック待ち（ステップ１９のＰＯＩＮＴ　Ｄ）でなるべく待ち状態とならないようにするためには、ステップ７～２０の処理にかかる時間Tb が優先するロボットのステップ１０～２５にかかる時間Ta よりも大きくなるようにしなければならない。つまり、追従するロボットが干渉域に達する前に優先するロボットがその干渉域を脱出するようにする。ところが時間Tb はＰＯＩＮＴ　Ｄの位置で決まってしまうので、なるべく優先するロボットのプログラムのステップ２４のＰＯＩＮＴ　Ｂの設定位置を上の方にしてTa の時間を少なくするようにする。したがって、Tb ＞Ta の範囲内でTB の時間が最大となるようにＰＯＩＮＴ　Ｂの位置を決定する。このように設定することができれば、追従するロボットの待ち時間をなくすることができる。

【００４８】つぎに、優先するロボットのプログラムのステップ２６でＰＯＩＮＴ　Ｂ＋１を固定した場合に、追従するロボットが優先するロボットと干渉しないかを、追従するロボットのプログラムのステップ２８におけるＰＯＩＮＴ　Ｅまでチェックする。つまり、ＰＯＩＮＴ　Ｅよりも前のステップにおいて、優先するロボットのプログラムのＰＯＩＮＴ　Ｂ＋１からＰＯＩＮＴ　Ｃまでと干渉しないかを確認する。この確認によって追従するロボットのプログラムのステップ１９からステップ２８までの処理に要する時間Ｔｄが求められる。優先するロボットのプログラムのステップ２５からステップ２９の処理に要する時間Ｔc も同様にして求められる。

【００４９】したがって、２台のロボットを同時に起動させたときのインターロック待ち時間は、次の４通りの式のいずれかによって求めることができる。

①　Ta ＜Tb かつTC ＜Td の場合には、追従するロボットの待ち時間は０。

②　Ta ＜Tb かつTC ＞Td の場合には、追従するロボットの待ち時間はTC －Td 。

③　Ta ＞Tb かつTC ＞Td の場合には、追従するロボットの待ち時間はTa －Tb 。

④　Ta ＞Tb かつTC ＞Td の場合には、追従するロボットの待ち時間はTa －Tb ＋TC －Td 。

【００５０】なお、ＰＯＩＮＴ　Ｂを最初のロジックで仮想的に決定したときに、①の条件を満たす位置（ステップ）ではなかった場合には、この仮に設定したそのＰＯＩＮＴ　Ｂの位置を１ステップづつ増やしていって、ＰＯＩＮＴ　Ｃの１ステップ前までについて繰り返し上記の待ち時間を計算し（①～④）、待ち時間を最小とするステップにＰＯＩＮＴ　Ｂ，ＰＯＩＮＴ　Ｅを設定する。

【００５１】以上がＳ１５のステップで行われる具体的な内容である。この処理は動作時間演算部２０、インターロック自動設定部１６、ロボットプログラム変換部１８によって分担して行われ、インターロック情報、インターロック解除情報、進入許可情報は最終的にロボットプログラム変換部１８によって付加されて、新ロボットプログラムとして外部装置に出力される。

【００５２】なお、以上の説明では、２台のロボットが干渉する場合を一例として述べたが、本発明は、これに限られず、複数台のロボットが相互に干渉する場合についても同様に適用可能であることはいうまでもない。

【図面の簡単な説明】

【図１】　本発明方法に使用するロボットの一例を示す図である。

【図２】　本発明方法を実施する装置を示す図である。

【図３】　本発明方法を示すフローチャートである。

【図４】　本発明方法の説明に供する図である。

【図５】　本発明方法の説明に供する図である。

【図６】　本発明方法の特に進入許可信号の付加の説明に供する図である。

【図７】　本発明方法の特に進入許可信号の付加の説明に供する図である。

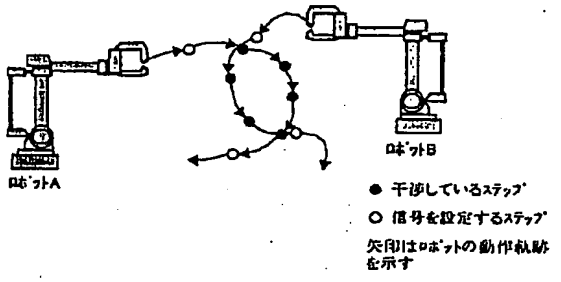
【符号の説明】

１０…ロボットプログラム記憶部、

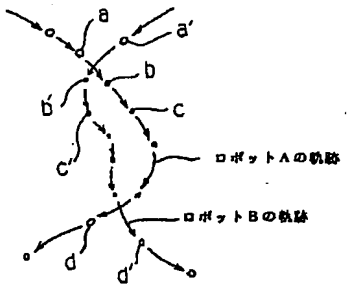
１２…シミュレーション実行部、

１４…干渉チェック部、
１６…インターロック自動設定部、
１８…ロボットプログラム変換部、
２０…動作時間演算部。

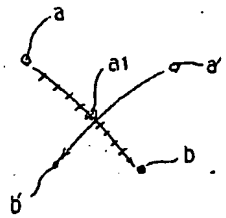
【図１】



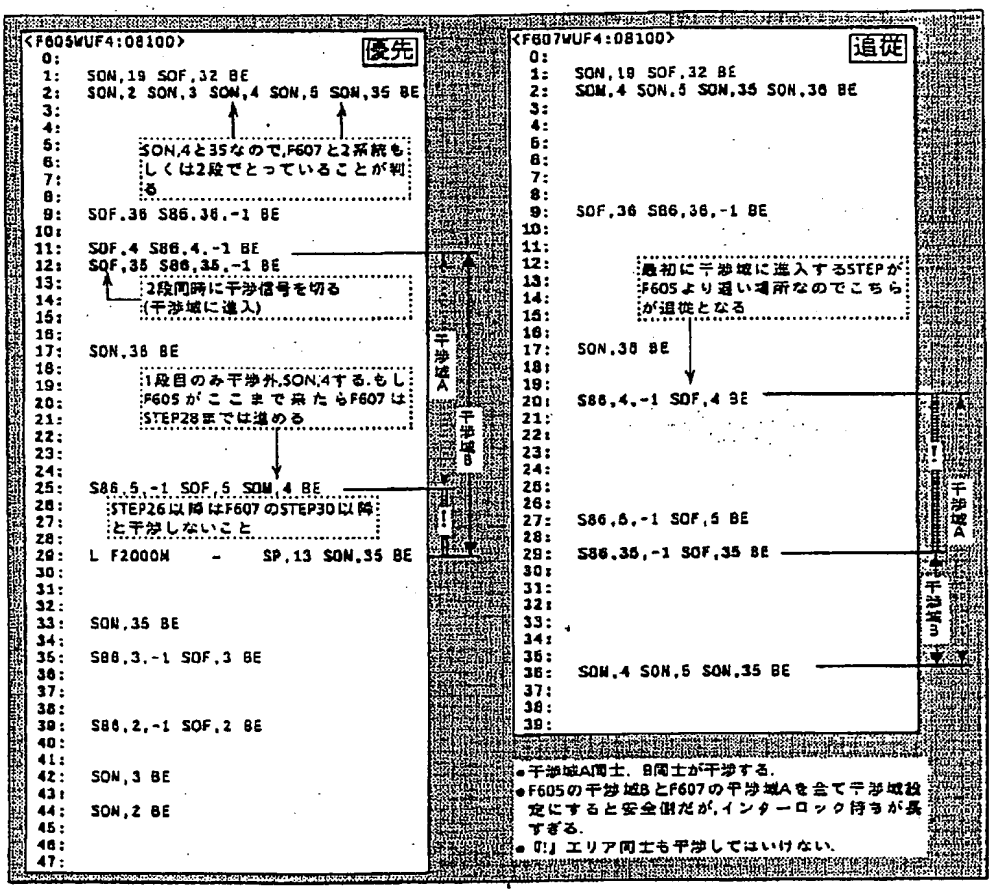
【図４】



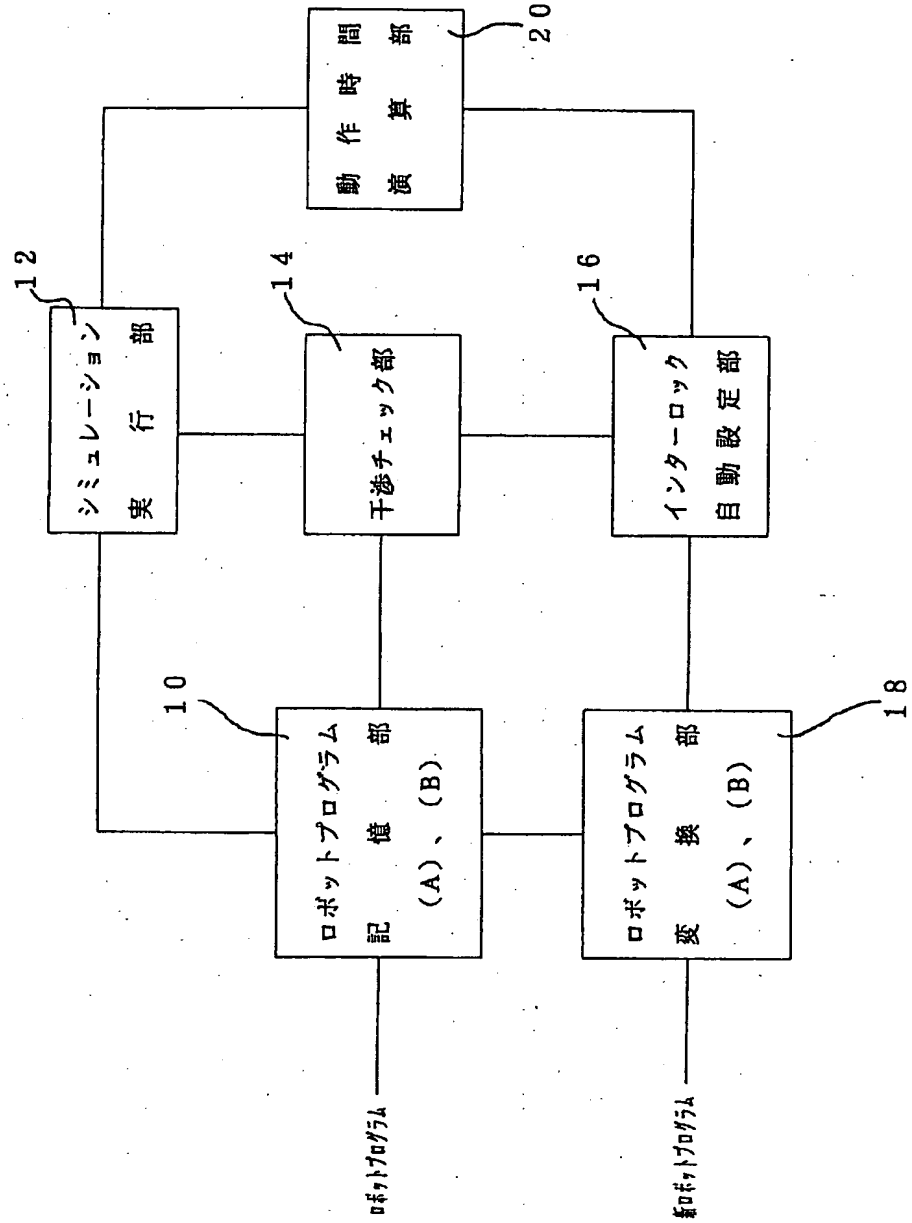
【図５】



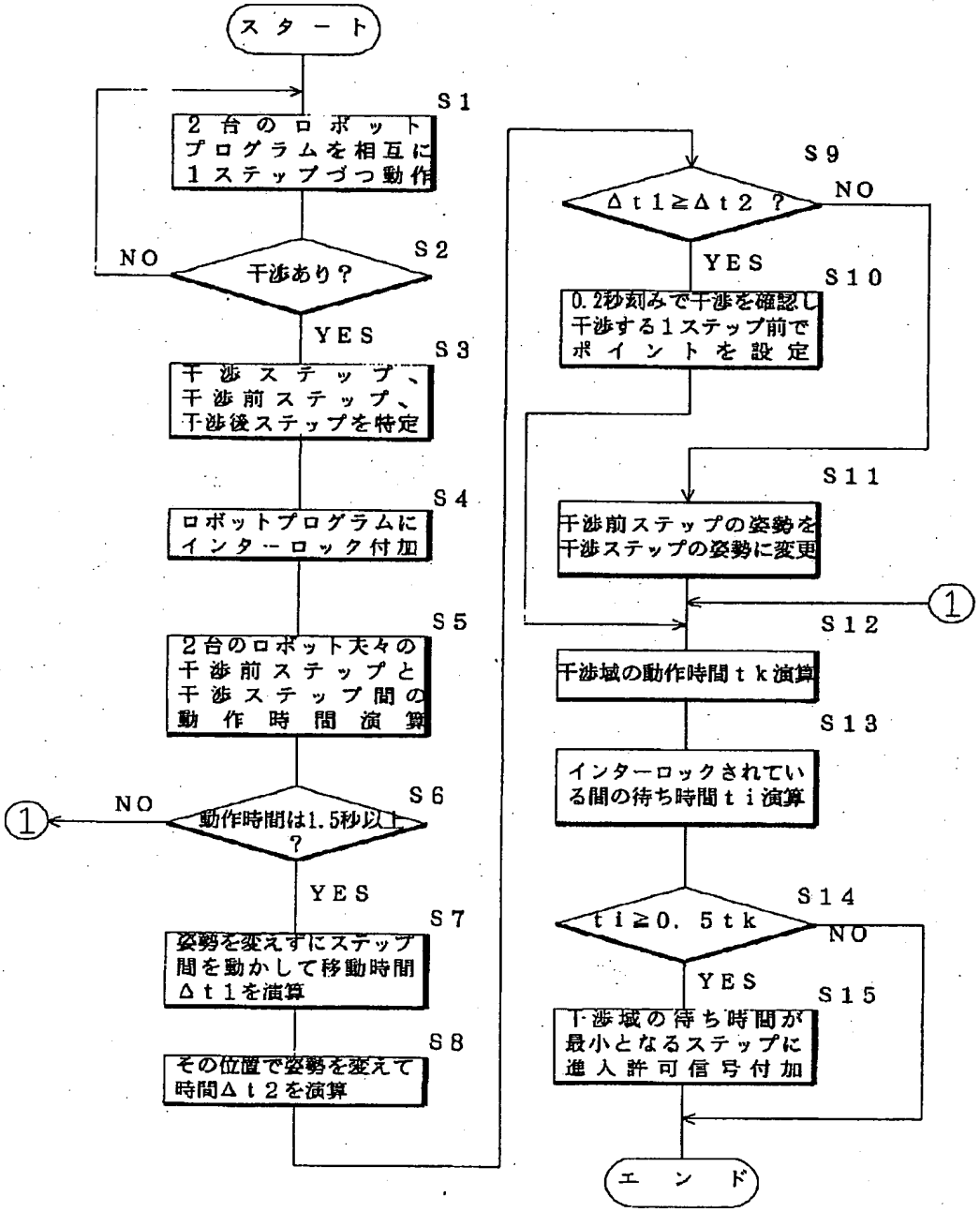
【図６】



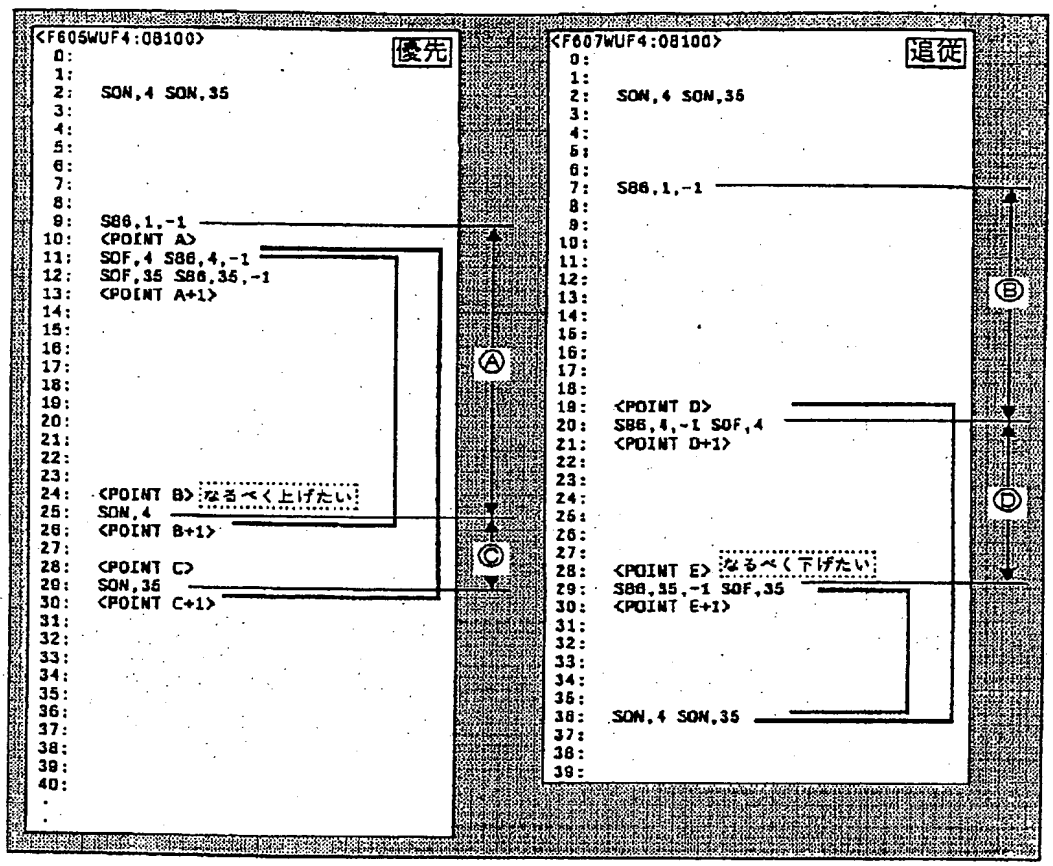
【図２】


動作時間演算部
20
シミュレーション実行部
12
干渉チェック部
14
インターロック自動設定部
16
ロボットプログラム記憶部（A）、（B）
10
ロボットプログラム変換部（A）、（B）
18
ロボットプログラム
新ロボットプログラム

【図３】


スタート
S1
２台のロボットプログラムを相互に１ステップづつ動作
S2
干渉あり？
NO
YES
S3
干渉ステップ、干渉前ステップ、干渉後ステップを特定
S4
ロボットプログラムにインターロック付加
S5
２台のロボット夫々の干渉前ステップと干渉ステップ間の動作時間演算
S6
動作時間は1.5秒以上？
NO
①
YES
S7
姿勢を変えずにステップ間を動かして移動時間Δｔ１を演算
S8
その位置で姿勢を変えて時間Δｔ２を演算
S9
Δｔ１≧Δｔ２？
NO
YES
S10
0.2秒刻みで干渉を確認し干渉する１ステップ前でポイントを設定
S11
干渉前ステップの姿勢を干渉ステップの姿勢に変更
①
S12
干渉域の動作時間ｔｋ演算
S13
インターロックされている間の待ち時間ｔｉ演算
S14
ｔｉ≧0.5ｔｋ
NO
YES
S15
干渉域の待ち時間が最小となるステップに進入許可信号付加
エンド

【図７】


<F605WUF4:08100>
優先
SON,4 SON,35
S86,1,-1
<POINT A>
SOF,4 S86,4,-1
SOF,35 S86,35,-1
<POINT A+1>
<POINT B>
なるべく上げたい
SON,4
<POINT B+1>
<POINT C>
SON,35
<POINT C+1>
Ⓐ
Ⓒ
<F607WUF4:08100>
追従
SON,4 SON,35
S86,1,-1
<POINT D>
S86,4,-1 SOF,4
<POINT D+1>
<POINT E>
なるべく下げたい
S86,35,-1 SOF,35
<POINT E+1>
SON,4 SON,35
Ⓑ
Ⓓ